WHAT IS CLAIMED IS:

1. A computer implemented method for simulating a system design containing at least two components, said method comprising:

identifying said components;

creating models of said components in a high level general purpose programming language;

creating a set of function calls in a high level general purpose programming language;

combining said models to form said virtual prototype;

linking one of said models with another of said models using said set of function calls;

executing said virtual prototype, wherein said models communicate and cycle accurate information is generated.

2. The computer implemented method of claim 1 wherein said creating models of said components step further comprises:

creating a blank component model;

adding a sub-component to said blank component model; and

configuring one or more parameters for said sub-component.

3. The computer implemented method of claim 2 wherein at least two of said sub-components are added to said blank component model, further comprising:

linking said sub-components with said set of function calls.

4. The computer implemented method of claim 2 wherein said components are added into a component repository after said components are created.

5. The computer implemented method of claim 1 wherein said set of function calls comprises:

a control interface for communications between the simulation environment and said components; and

a peer interface for communications between said components.

6. The computer implemented method of claim 5 wherein said peer interface comprises clock functions, access functions, and signal functions.

7. The computer implemented method of claim 1 wherein said combining step further comprises:

loading a component from a component repository;

identifying sub-components of said component; and

elaborating said sub-components.

8. The computer implemented method of claim 7 wherein said elaborating step further comprises:

instantiating said sub-components;

configuring said sub-components; and

linking said sub-components with said set of function calls.

9. The computer implemented method of claim 1 wherein said generating cycle accurate information step further comprises:

dividing activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

computing said first plurality of activities comprising said execute phase at a clock edge;

updating at said clock edge a state of said simulation environment; and

computing said second plurality of activities comprising said update phase at said clock edge.

10. A computer implemented method for simulating a digital system design in a cycle based simulation environment, comprising:

creating a system design model in a high level general purpose programming language, comprising at least two components;

creating a software interface in a high level general purpose programming language that is available to said at least two components;

executing said system design model, wherein said at least two components communicate via said interface; and

maintaining cycle accurate information during the simulation.

11. The computer implemented method of claim 10 wherein said creating a system design model step further comprises:

creating a blank component model;

adding a sub-component to said blank component model; and

configuring one or more parameters for said sub-component.

12. The computer implemented method of claim 11 wherein at least two of said sub-components are added to said blank component model, further comprising:

linking said sub-components with said set of function calls.

13. The computer implemented method of claim 11 wherein said components are added into a component repository after said components are created.

14. The computer implemented method of claim 12 wherein said set of function calls comprises:

a control interface for communications between said simulation environment and said components; and

a peer interface for communications between said components and between said sub-components.

15. The computer implemented method of claim 14 wherein said peer interface comprises clock functions, access functions, and signal functions.

16. The computer implemented method of claim 10 wherein said maintaining cycle accurate information step further comprises:

dividing simulation activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

computing said first plurality of activities comprising said execute phase at a clock edge;

updating at said clock edge a state of said simulation environment; and

computing said second plurality of activities comprising said update phase at said clock edge.

17. A computer program product for simulating a system design within a cycle based simulation environment, said computer program product comprising:

instructions for identifying at least two components in said system design;

instructions for integrating models of said at least two components, wherein said models are created in a high level general purpose programming language;

instructions for linking said models using a set of function calls created in a high level general purpose programming language; and

instructions for executing said system design, wherein said models communicate through said link and cycle accurate information is generated.

18. The computer program product of claim 17 wherein said instructions for creating models further comprises:

instructions for creating a blank component model;

instructions for adding at least one sub-component to said blank component model;

instructions for configuring one or more parameters for said sub-component; and

instructions for linking said sub-components using said set of function calls.

19. The computer program product of claim 18 further comprising instructions for adding said models of said components into a component repository after said models are created.

20. The computer program product of claim 18 wherein said instructions for creating a set of function calls comprise:

instructions for creating a control interface for communications between said simulation environment and said components; and

instructions for creating a peer interface for communications between said components and between said sub-components.

21. The computer program product of claim 20 wherein said instructions for creating a peer interface further comprise:

instructions for creating clock functions;

instructions for creating access functions; and

instructions for creating signal functions.

22. The computer program product of claim 17 wherein said instructions for generating cycle accurate information further comprise:

instructions for dividing simulation activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

instructions for computing said first plurality of activities comprising said execute phase at a clock edge;

instructions for updating at said clock edge a state of said simulation environment; and

instructions for computing said second plurality of activities comprising said update phase at said clock edge.

23. A method for simulating a design containing at least two components, said method comprising:

creating a model representing each of said at least two components, wherein said models correspond via at least one function call and comprise a virtual prototype; and

executing said virtual prototype to generate cycle accurate information.

24. The method of claim 23 wherein said creating step comprises creating said model in a high level general purpose programming language.

25. The method of claim 23 wherein said creating models of said components step further comprises:

creating a blank component model;

adding a sub-component to said blank component model; and

configuring one or more parameters for said sub-component.

26. The method of claim 25 wherein at least two of said sub-components are added to said blank component model, further comprising:

linking said sub-components with said at least one function call.

27. The method of claim 25 wherein said components are added into a component repository after said components are created.

28. The method of claim 23 wherein said executing step is performed in a cycle based simulation environment, said method further comprising:

creating a set of function calls in a high level general purpose programming language, wherein said set of function calls comprises:

a control interface for communications between said cycle based simulation environment and said components; and

a peer interface for communications between said components.

29. The method of claim 28 wherein said peer interface comprises clock functions, access functions, and signal functions.

30. The method of claim 23 wherein said virtual prototype is created by steps comprising:

loading a component from a component repository;

identifying sub-components of said component; and

elaborating said sub-components.

31. The method of claim 30 wherein said elaborating step further comprises:

instantiating said sub-components;

configuring said sub-components; and

linking said sub-components with said set of function calls.

32. The method of claim 23 wherein said executing step is performed in a cycle based simulation environment and said generating cycle accurate information step further comprises:

dividing activities in said cycle based simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

computing said first plurality of activities comprising said execute phase at a clock edge;

updating at said clock edge a state of said simulation environment; and

computing said second plurality of activities comprising said update phase at said clock edge.

33. A computer program product for simulating a system design containing at least two components comprising:

instructions for integrating a model representing each of said at least two components, wherein said models communicate via at least one function call and comprise a virtual prototype; and

instructions for executing said virtual prototype to generate cycle accurate information.

34. The computer program product of claim 33 wherein said models are created in a high level general purpose programming language.

35. The computer program product of claim 33 wherein said instructions for executing said virtual prototype to generate cycle accurate information further comprise:

instructions for dividing simulation activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

instructions for computing said first plurality of activities comprising said execute phase at a clock edge;

instructions for updating at said clock edge a state of said simulation environment; and

instructions for computing said second plurality of activities comprising said update phase at said clock edge.